The Computational Engineering and Lightweight Structures Website

Name: Patrick Highland _____
Date: 8/16/02
High School: Bel Air High School
Laboratory: ARL - CISD
Mentor: Dale Shires _____

Abstract

A PHP web portal system called "PHP Nuke" was used as a dynamic engine to set up the Computational Engineering and Lightweight Structures website. Through a combination of pre-designed and customized modules, the website was able to help organize various projects within the Computational Engineering and Lightweight Structures team. The web portal system provided easy to use dynamic scripts for updating the website's news, downloads, and specially designed bug reporting and tracking system.

Several scientific computing codes are under development in the United States Army Research Laboratory's Computational and Informational Sciences Directorate. One group of researchers task was to develop computer programs that would be used for computational engineering and lightweight structures development. These programs are called COMPOSE (Composite Manufacturing Process Simulation Environment) and DINOSAURUS (Dynamic Integration Operators for Structural Analysis Using Robust Unified Schemes). The team needed a way to offer support to their customers while at the same time getting feedback form them on how to improve the program. A way for the customer to report program bugs and a way for an administrator to classify, store, and resolve the bugs was also desired. The simplest solution to all of these needs was a comprehensive and dynamic website.

There are many ways to develop a website. Each way has its advantages and disadvantages. It was decided that the website would need large database support in order to hold the large amount of clientele information and all of the reported errors from the not yet developed bug tracking system. One largely supported package for large databases is MySQL. MySQL is open source and perfect for customization and precision tracking under large stress loads. However, in order to connect to the database through the Internet you must use a scripting language of some sort. There were many options including Microsoft's Active Server Pages (ASP), Java Server Pages (JSP), Perl, and PHP. Perl (also known as CGI) is still considered the industry standard. It is a C based language, which makes it very powerful. PHP is a newer language. It is also C-based but much easier to program and is more powerful in that it can generate HTML code and images on the fly giving it an edge over Perl. We also chose PHP because many web developers believe it will be the next industry standard.

The next dilemma came in how to design the website using PHP. Since the

Internet has become available their has been information sharing around the world like never before in human history. Many people and companies distribute free source code to offer you a head start in developing applications and websites. Since there is no point in reinventing the wheel it was decided to use a base web site engine. I decided to use a web portal system name "PHP Nuke" which is free software released under the GNU/GPL license. PHP-Nuke is a news-automated system specially designed to be used in Intranets and on the Internet. The Administrator has total control of his web site, registered users, and he will have in the hand a powerful assembly of tools to maintain an active and 100% interactive web site using databases.

A great advantage to using PHP Nuke's web portal system over others is how easily it can be customized, upgraded, and maintained. The administrator can custom-design modules that can be uploaded to the site, tested, evaluated, and then implemented. Also there is almost no website maintenance at all. Once you are done designing all of the modules there is never any need to go into the source code again. Through a collection of powerful PUP scripts things like news, downloads, Internet links, and opinions can all be uploaded to the site, sorted, and stored for future viewing without editing any PHP or HTML.

There was a lot of customization that went into the designing of the Computational Engineering and Lightweight Structures (CELS) website. All of the images that came with PHP Nuke had to be redone. I used Adobe Photoshop to do this. Also, a lot of the index source code had to be rewritten to offer the functionality the customer needed to accomplish the certain tasks outlined by the CELS team. These criteria were organized easy post news, organized downloads, bug tracking for COMPOSE and DINOSAURUS, search features, password protection, submittable web

links, and easy maintenance as a whole. All of these needs were satisfied.

The password protection was implemented in three ways. There are three security levels. The first, anonymous users who can browse some of the websites information but cannot view many things such as the downloads. They can also not submit error reports. Second, there are registered users who are registered at the administrator's digression. Registered users can browse the site, download files, and submit error reports. Finally, the highest security level is an administrator. Inside the administrator level there are multiple security branches that can be enabled by the preeminent administrator. This means that there can be multiple administrators but not all of them necessarily have complete access to the website. The super administrator can basically do anything possible. He can even view and edit the source code to the website.

Most of the website's development time was spent on the error reporting modules for DINOSAURUS and COMPOSE. The CELS team wanted a set of modules to track errors in their programs. I decided to have two separate modules, a client side and an administrator side. Inside each module would be two functions. One for COMPOSE and one for DINOSAURUS. Both modules were written in PHP. The client side contained five fields: a unique id, a submission date, a user id, error number, and a short description of the error. The unique id and submission date are generated on the fly by the server. Once the fields are submitted by the user he may not edit them in any way or delete them. The administration side has twelve fields: a unique id, a submission date, a user id, error number, a short description of the error, a class (software or documentation), a status (open, closed, analyzed, suspended), a severity (non-critical, serious, critical), a priority (low, medium, high), a maintainer id, a resolution, and a resolution date. The unique id, submission date, user id, error number, and short description were submitted by the client

on the user side but are editable by the administrator. The administrator submits the class, status, severity, priority, maintainer, and resolution. The server generates the resolution date on the fly.

Once the errors are classified and stored in the MySQL database by the two PHP script modules, it is time to analyze them. The maintainer or administrator may sort any or all of the fields by any other one field. By doing this the administrator or maintainer can see patterns in the errors. Noticing patterns will help more efficiently identifying and tracking recurring problems in the program and hence the source code. The error tracking modules offer the CELS team the power to more efficiently cater to their client's needs with better code and support.

Some features that may be added by another member of the CELS team in the future were automatic login detecting for the client side script and automatic error recording. In even future versions it is hoped that the COMPOSE and DINOSAAURUS programs will be able to themselves detect all the information that needs to be sent to the client and administration sides such as who the program is registered to and the exact source of the error inside of the program. The actual modules for processing the data that have already been written should stay useful with little or no maintenance for quite some time. One last feature that is under consideration is a VRML environment. VRML is an online language that models three-dimensional environments. Clients would be able to view complex three-dimensional environments and models without having to purchase an expensive viewing program.

The next step in the website's development was the transferring of the website form the development server to a server accessible from the outside world. Once the

website was transferred the process of beta testing the site began. A few minor bugs were found but were easily corrected. All modules worked successfully and performed their intended functions.

The Computational Engineering and Lightweight Structures website was a success. It provided a web interface for their clients and an outlet to actively promote and support their research. Through a combination of modules written by myself and given to me in the PHP Nuke Web Portal System, the needs (organized easy post news, organized downloads, bug tracking for COMPOSE and DINOSAURUS, search features, password protection, submittable web links, and easy maintenance as a whole) were all met. The site can easily be upgraded by an administrator at any time and can be maintained without ever going into the source code. PHP should remain a good scripting language for the years to come and take the lead over Perl as the industry standard for web scripting and database interface. The fact that all website data is stored in MySQL in over 75 tables means that the website engine has a fast, reliable, and precise way of accessing data. The database is easily backed up and is much safer and faster way than conventional ways of storing. The website should have a long life with proper maintenance.

Bibliography

Meloni, Julie. PHP Essentials. Prima Publishing: Rocklin, 2000.

Acknowledgements

I would like to thank my mentor, Dale Shires, for accepting me as his intern and taking the time to pass his valuable knowledge to me.

I would like to thank Brady Bellinger for his help in uploading the CELS website to multiple servers and his help in debugging the website.

I would like to thank Ann Brodine for her time and effort that she put into the SEAP program.

I would like to thank Dr. Jon Brand for organizing tours and talks that were useful and enjoyable.